Covering cover pebbling number of products of paths

A THESIS
SUBMITTED TO THE FACULTY OF THE
UNIVERSITY OF MINNESOTA
BY

Ian McGahan

IN PARTIAL FULFILLMENT OF THE REQUIERMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Dalibor Froncek, Sylwia Cichacz-Przenioslo

May 2015

Abstract

There are a variety of pebbling numbers, such as classical pebbling number, cover pebbling number, and covering cover pebbling number. In this paper we determine the covering cover pebbling number for Cartesian products of paths. The covering cover pebbling number of a graph, G, is the smallest number of pebbles, n, required such that any distribution of n pebbles onto the vertices of G can be, through a sequence of pebbling moves, redistributed so that C, a vertex cover of G, is pebbled. Traditionally, a pebbling move is defined as the removal of two pebbles from one vertex and the placement of one pebble on an adjacent vertex. In this paper we provide an alternative proof for the covering cover pebbling number of cycles to the proof given in [9] and prove that the covering cover pebbling number for a Cartesian product of paths is given by the following:

$$\sigma(P_n \square P_m) = \begin{cases} \frac{4}{9}(2^m - 1)(2^n - 1) & \text{when } n, m \equiv 0 \pmod 2 \\ \frac{1}{9}(2^{2+m+n} - 5(2^m) - 5(2^n) + 4) & \text{when } n, m \equiv 1 \pmod 2 \\ \frac{1}{9}((2^{2+m} - 5)(2^n - 1) - 3m2^n) & \text{when } n \equiv 0 \text{ and } m \equiv 1 \pmod 2 \end{cases}$$

Contents

List of Figures

1 Introduction

1.1 The Origins of Graph Theory

Graph Theory has its origins in what is today known as the Königsberg Bridge Problem, a problem that serves as an excellent example to outline the building blocks of graph theory. Königsberg, now Kaliningrad in Russia, was a city located on the Pregel River. The city spanned both banks as well as two large islands. The parts of the city were connected by a total of seven bridges. As the story goes, the citizens of Königsberg wondered if it was possible to leave home in the morning and, by walking through the city, cross every bridge exactly once to return home in the evening. The only way to cross any part of the Pregel River was via one of the seven bridges and each time a bridge was crossed it must be crossed completely. For the sake of the problem crossing a bridge halfway and then turning back did not count. The approximate geometry for the problem is given in Figure 1. The bridges have been labeled a, b, c d, e, f, and g, the banks of the river are B and C, and the islands are A and D.

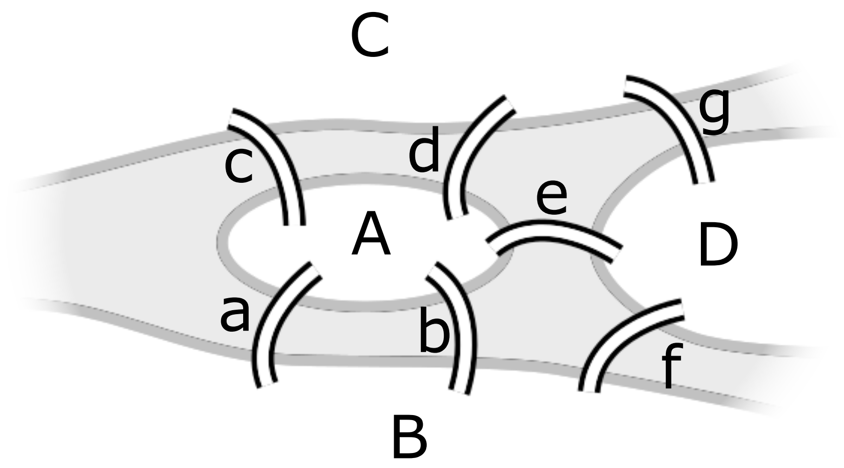

Figure 1: Königsberg Bridge Problem

Leonhard Euler proved the task to be impossible in 1735, and in doing so laid the foundations for graph theory. Euler argued that the starting point did not matter. If one could start at one point, say A, cross every bridge, and return to A then it follows that following the same path from a different starting point, say B, would result in a solution as well. This meant that the actual geography of the town was irrelevant, reducing the problem to four locations and seven bridges connecting them. This is abstraction provides the basis for how we define a graph. A *graph*, G, is made up of three things: a set of vertices $V(G)$, a set of edges $E(G)$, and a relation that associates each edge with two (not necessarily distinct) vertices. The two vertices associated with a given edge $e \in E(G)$ are called the *endpoints* of e and are said to be *adjacent*. In the context of the bridge problem, the vertex set is the set of land masses A, B, C, and D. The edge set is the set of bridges a, b, c, d, e, f, and g. The endpoints of each edge is simply the two land masses that each bridge connects. For example the vertices associated with the edge a are B and A. We can then represent the Königsberg bridge problem the graph in Figure 2.

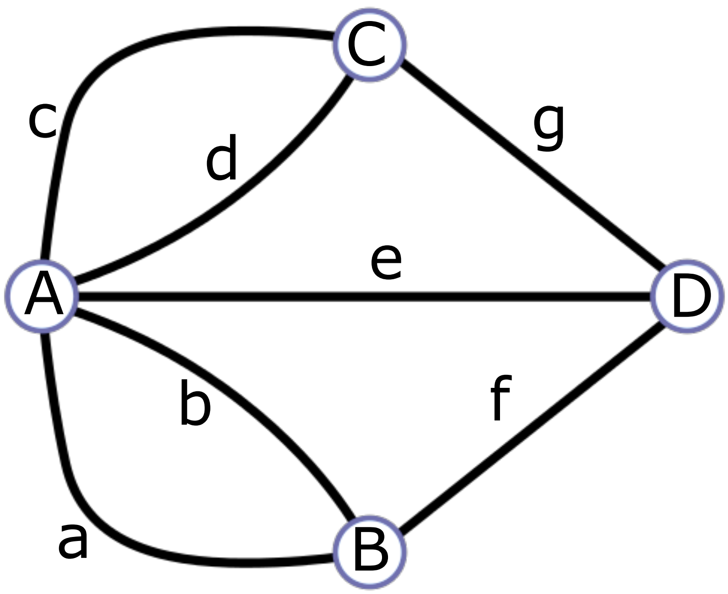

Figure 2: Bridge Problem graph

It is important to note here that the distance between vertices, or their relative location in two-space are completely irrelevant. All we are concerned with is whether or not there is an edge between two vertices. We also notice that some of the vertices in our Bridge Problem graph have more than one edge between them. For example, B and A have two edges associated with them. If a graph has two vertices with more than one edge between them these edges are called *multiple edges*. In our definition of a graph we mentioned that the relation does not need to associate unique vertices with an edge. If an edge has the same vertex as both of its endpoints this edge is called a *loop*. Our Bridge Problem graph does not have any loops. A graph which does not have any loops or multiple edges is called a *simple graph*. The graphs that we examine in this paper are all simple graphs because in pebbling we are only interested in whether we can get from one vertex to another, not whether there are multiple ways to do this. Another thing we can notice about our problem is that there is no island sitting alone without a bridge to it. In our graph we would represent this as a vertex without any edges associated with it. We would call such a vertex an *isolated vertex*. We can generalize this idea of an isolated vertex to a collection of vertices. Two vertices, u and v, are *connected* if there is a series of vertices $u, v_1, \ldots, v_j, v$ so that u is adjacent to v_1, v is adjacent to v_j and v_i is adjacent to v_{i+1} for each i. A set of vertices is *connected* if any pair of vertices in the set is connected. Much like how in the Bridge Problem we can get from any landmass to any other landmass by crossing some bridges. We call a set of vertices which are connected a *connected component* of G. A graph consisting of a single connected component is a *connected graph*. The graphs we examine are all connected because, as we will see later, disconnected graphs are trivially impossible to pebble.

In the Königsberg bridge problem we wanted to walk over each bridge exactly once. In graph theoretic terms, we need a trail that passes through every edge of G, our Bridge Problem graph. A *walk* is a list $v_0, e_1, v_1, e_2, \ldots, e_n, v_n$ where each v_i is a vertex and each e_i is an edge and the endpoints of e_i are v_{i-1} and v_i. A *trail* is a walk which never repeats an edge. We say a trail is *closed* if v_0 and v_n are the same vertex. If a graph has a closed trail that goes through every edge we say the graph is *Eulerian* and the closed trail is an *Eulerian circuit.* A *path* is a walk that never repeats a vertex. Note that out of necessity if a path does not repeat a vertex it can never repeat an edge either. A *(u,v)-path* is a path whose first and last vertices are u and v. The other vertices are called *internal vertices.* We use the same notation of a (u, v)-walk, or trail to denote a walk or trail that begins at u and ends at v.

In the context of the Königsberg bridge problem, there will be a way to cross every bridge exactly once if and only if the Bridge Problem graph has a closed Eulerian trail. Euler observed that each time we enter and leave a landmass we use two bridges. Further, the first and last bridge we cross, if we could start and end on the same landmass, would also be paired. Thus if there is to be a closed Eulerian trail of our Bridge Problem graph, there must be an even number of bridges leaving each landmass (or entering). However, each vertex has an odd number of edges, so each landmass has an odd number of bridges. Therefore, there is no way to wake up in Königsberg, go for a walk in the city, cross every bridge exactly once, and in doing so return home. This argument was generalized to the Theorem that says if a graph has a vertex of odd degree it cannot have a closed Eulerian trail.

1.2 Families of graphs

In this paper we use several families of graphs. The first family of graph we discuss is the family of paths. We have already introduced the concept of a path within another graph. A *path* is a graph of the form $v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n$ where each v_i is a distinct vertex and each e_i is an edge with endpoints v_i and v_{i+1}. We call the vertices v_1 and v_n the *endpoints* of the path. In this paper we denote such a path by P_n. Note that n is the number of vertices and one more than the number of edges. For our investigation paths are an excellent family of graphs to start with because, in pebbling, we are very interested in distances between two vertices. For two vertices u and v we say the *distance* between them, denoted by $d(u, v)$, is given by the number of edges in the shortest path between u and v. In a path, P_n, there is only one path between any pair of vertices so the shortest path is easy to calculate. For example, the path P_6 is shown in Figure 3. Notice that the distance between the endpoints is 5, and that there is only one path between them.

Figure 3: Path P_6

A second family of graphs we discuss is the family of cycles. A *cycle* is a path with an additional edge between v_1 and v_n and is denoted by C_n. Note here that the number of vertices and the number of edges is the same. For example, the cycle C_8 is shown in Figure 4. Notice that C_8 has eight edges and eight vertices. In a cycle there are two paths between any pair of vertices.

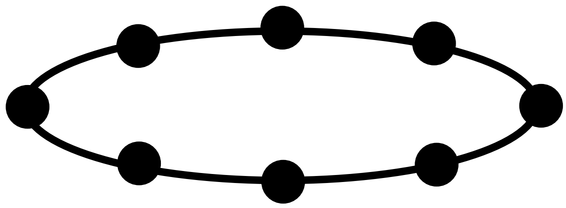

Figure 4: Cycle C_8

In order to define our third family of graphs we need to define the notion of Cartesian product. Formally, the *Cartesian product* of two graphs G and H, denoted by $G\Box H$, has the vertex set $V(G) \times V(H)$ and the vertex (u, v) is adjacent to (u', v') if and only if $u = u'$ and there is an edge between v and v' in H or $v = v'$ and there is an edge between u and u' in G. Another way to think of this is that we replace each vertex in G with a copy of H and then add an edge between a pair of vertices in different copies of H if they are the same relative vertex in H and those copies of H replace adjacent vertices in G. For example, the graphs P_3, P_4, and $P_3\Box P_4$ are shown in Figure 5. Notice that $P_3\Box P_4$ and $P_4\Box P_3$ are just the same graph, just represented differently in space. However, since we are not concerned with the spatial representation of graphs, we say that these two graphs are the same, or *isomorphic*.

The third family of graphs we discuss are Cartesian products of paths. That is for a path of length n and a path of length m the Cartesian product is denoted by $P_n \square P_m$ and follows directly from the definition given above. An example is given in Figure 5.

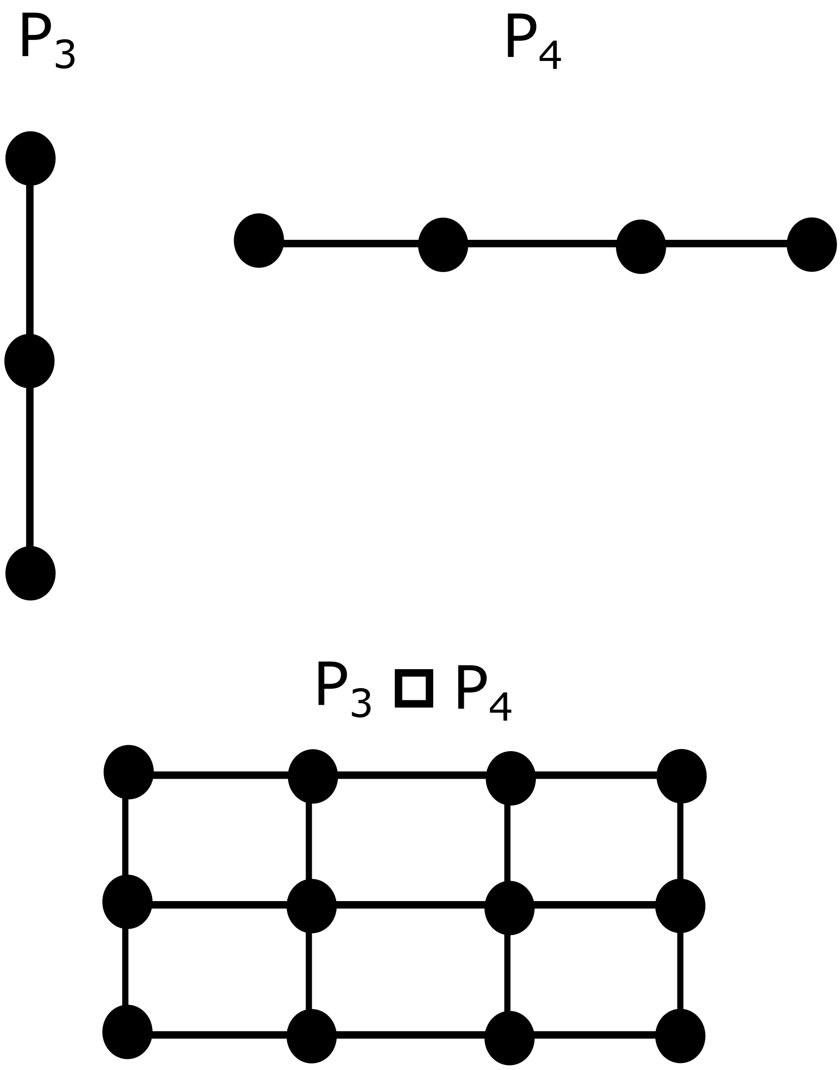

Figure 5: P_3, P_4, and $P_3 \square P_4$

The three families of graphs discussed so far comprise the set of graphs for which we compute or use known results for the covering cover pebbling number. However it is useful here to define three other families of graphs as they are mentioned in this paper as well. First, the complete graph, denoted by K_n, is a simple graph on n vertices so that every vertex is adjacent to every other vertex. For example, the complete graph on five vertices, K_5, is shown in Figure 6.

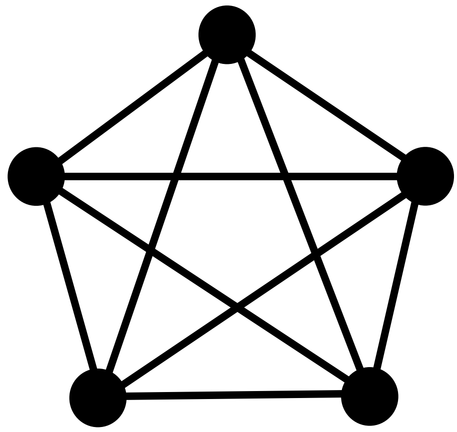

Figure 6: Complete graph K_5

Second is a class of graphs called bipartite graphs. A bipartite graph G has a vertex set $V(G)$ which can be divided into two disjoint subsets A and B such that the only edges in G have an endpoint in A and an endpoint in B. Namely, there are no edges between vertices in A and no edges between vertices in B. It is important to note that the two partite sets do not need to have the same number of vertices. In Figure 7 the graph named G is an example of a bipartite graph where the partite sets have size three and four respectively. One particular subset of bipartite graphs are complete bipartite graphs and are denoted by $K_{m,n}$. Here the vertex set $V(G)$ is divided into disjoint sets A and B where A has m vertices and B has n vertices. Further, for any vertices $v \in A$ and $u \in B$ there is an edge from u to v. The graph named $K_{3,4}$ in Figure 7 is an example of a complete bipartite graph.

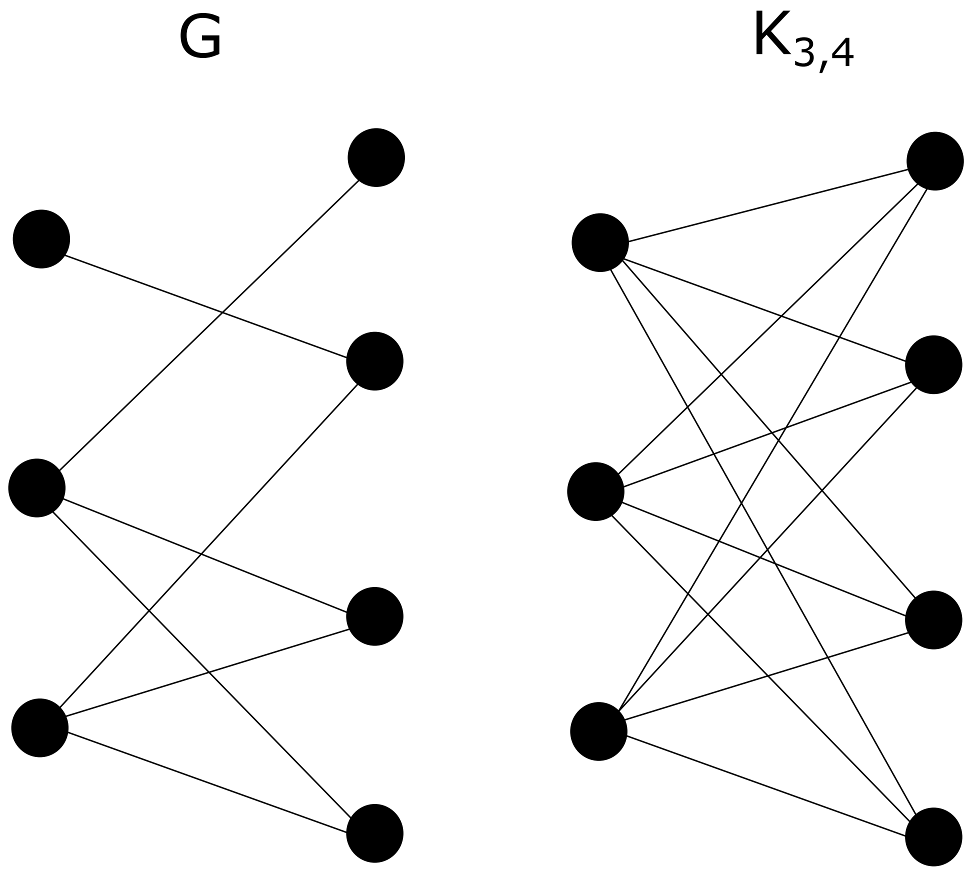

Figure 7: Bipartite graph and $K_{3,4}$

1.3 Graph Pebbling

Graph pebbling, first proposed by Lagarias and Saks, was conceived as a tool to prove a number theoretical conjecture of Erdős [3]. The conjecture is shown in Theorem 1.1. F. Chung proved the following strengthened version of this conjecture using graph pebbling in [1] in 1989, and in doing so formalized the idea of graph pebbling into writing.

Theorem 1.1 [3]. *For any positive integer n, every sequence $(a_k)_{k=1}^{n}$ of n integers contains a nonempty subsequence (a_k) with $k \in K$ such that $\sum_{k \in K} a_k \equiv 0 \pmod{n}$ and $\sum_{k \in K} \gcd(a_k, n) \leq n$*

Since then there have been a number of results in graph pebbling as well as the introduction of multiple variations of graph pebbling. The most natural explanation of pebbling is in the form of a game. The graph pebbling game is played by two players, *Alice* and *Beth* on a simple connected graph G. It is worth noting here that the specific names *Alice* and *Beth* are not required names for players of this game, merely examples. First, Alice gives Beth some number of pebbles which Beth then distributes onto the vertices of G. We will refer to this initial distribution of pebbles by Beth as an *initial configuration*. Then Alice may make pebbling moves until a winning configuration is achieved. A *winning configuration* is a configuration of pebbles which meets a win condition specific to the variant of the pebbling game being played. The goal for Beth is to give Alice an initial configuration from which a winning configuration cannot be achieved. A *pebbling move* consists of removing two pebbles from the same vertex and then adding a single pebble to an adjacent vertex. When we say *configuration* we refer to a distribution of pebbles on the vertices of G at any point during the game. Essentially a configuration is a specific game state.

In the target pebbling variation a vertex, v, is specified prior to Alice selecting the number of pebbles which Beth distributes. The winning configuration is any configuration in which v has at least one pebble on it. The *pebbling number* of a graph G is the minimum number of pebbles required so that Alice can reach the winning configuration from any initial configuration and for any target. The pebbling number is traditionally denoted by $\pi(G)$. G. Hurlbert published a survey of pebbling results in [4]. The pebbling number, $\pi(G)$, is also referred to as the target pebbling number or traditional pebbling number to distinguish it from other pebbling variants because this is the original version of the problem.

In the cover pebbling variation the game play is the same but the winning configuration, instead of a target vertex, is a configuration where every vertex of G has at least one pebble on it. The *cover pebbling number* of a graph G is the minimum number of pebbles required so that from any initial configuration the winning configuration can be reached. The cover pebbling number is typically denoted by $\gamma(G)$. Crull et. al.

determine the cover pebbling number of complete graphs, paths, and trees in [the cover pebbling number of graphs] and in [5] G Hurlbert and B. Munyan determine the cover pebbling number of hypercubes.

Before discussing a third variation of pebbling we need to define a vertex cover. A *vertex cover* is a subset C of $V(G)$, so that every edge in $E(G)$ has at least one endpoint in C. In Figure 8 we exhibit two possible vertex covers of a graph called the bow tie. The black vertices represent the set C in each of the two examples. Both covers are also examples of *minimal covers* because removal of one vertex from the cover results in the set no longer being a vertex cover. The cover on the left has an additional property in that it is an example of a minimum cover. A *minimum cover* is a vertex cover of G using a minimum number of vertices to achieve the property of being a cover. Although both covers are minimal, the one on the left uses fewer vertices and, as it turns out, there is no smaller set of vertices which has the property of being a cover. Thus the cover on the left is minimum.

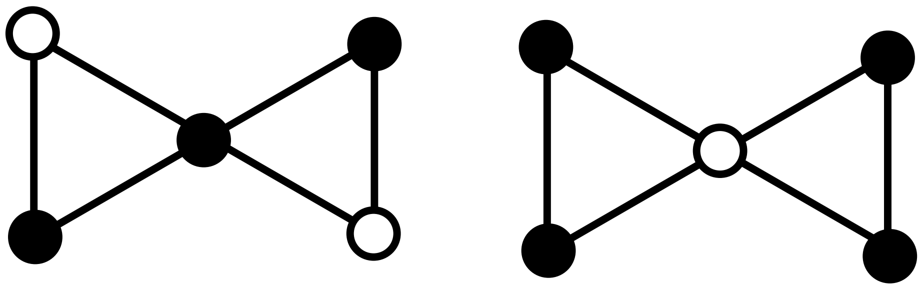

Figure 8: Vertex covers of the bow tie

The covering cover pebbling variation on graph pebbling was introduced by A. Lourdusamy and A. Tharai in [6]. In the covering cover pebbling variant the winning configuration is any configuration where the set of vertices with pebbles on them forms a vertex cover of G. The minimum number of pebbles required such that from any initial configuration a winning configuration is attainable is the *covering cover pebbling number* of G and is typically denoted by $\sigma(G)$. The covering cover pebbling number for many families of graphs is already known. In [6], Lourdusamy and Tharani determine the covering cover pebbling number for complete graphs, paths (given in Theorem 1.2), wheels, complete r-partite graphs, and binary trees.

Theorem 1.2 [6]**.** *The covering cover pebbling number for paths is given by* $\sigma(P_n) = \left\lfloor \frac{1}{3}(2^n - 1) \right\rfloor$.

Lourdusamy and Mathivanan determine the covering cover pebbling number for cycles in [9] (given in Theorem 1.3), the square of a path in [8] and the square of a cycle in [7]. In the next section we develop the necessary tools to determine the covering cover pebbling number for Cartesian products of paths.

Theorem 1.3 [9]**.** *The covering cover pebbling number for a cycle is given by*

$$\sigma(C_n) = \begin{cases} \left\lceil \frac{2^{k+2}-5}{3} \right\rceil & m = 2k (k \geq 2) \\ 2^k - 1 & m = 2k-1 (k \geq 2) \end{cases}$$

2 Results

2.1 Paths

In order to determine the covering cover pebbling number for products of paths we develop several necessary lemmas and theorems. First we show that the covering cover pebbling number of a path is realized by the initial configuration in which all pebbles are placed on an end vertex of the path and that any other initial configuration requires fewer pebbles. We follow this up with an analogs result for cycles, and in doing so provide an alternative proof to the proof given in [9] by Lourdusamy et. al. for the covering cover pebbling number of a cycle of even length.

Theorem 2.1. *Let P_n be a path of length n and let ϕ be an initial configuration of $\sigma(P_n) - 1$ pebbles onto P_n that is not every pebble on the same end vertex. It is possible to pebble a cover of P_n from any such ϕ.*

Proof. Let P_n be a path with $V(P_n) = \{v_1, v_2, \ldots, v_n\}$. Now let α be an initial configuration of pebbles on P_n such that not every pebble is located on the same end vertex, (i.e. either there are pebbles on both v_1 and v_n or there is at least one pebble on v_i

where $1 < i < n$). Further suppose that every pebble is necessary to successfully pebble a cover of P_n. In other words, it is impossible to reach a winning configuration from α', where α' is α with an arbitrary pebble removed. In order to pebble a cover of P_n we must pebble either v_{n-1} or v_n. From the proof Theorem 1.2 given by Lourdusamy and Tharani in [6] we know that if all pebbles are placed on v_1 we need $\left\lfloor \frac{1}{3}(2^n - 1) \right\rfloor$ pebbles, and of these there are 2^{n-1} which are for pebbling v_{n-1}. Now if there is at least one pebble on v_n we need not pebble v_{n-1} at all. Thus there is no vertex we need to pebble costing 2^{n-1} so we do not need $\left\lfloor \frac{1}{3}(2^n - 1) \right\rfloor$ pebbles. If there is no pebble on v_n then it follows that there is at least one pebble between v_1 and v_{n-1}. If the pebble lies on v_{n-1} then we still do not need to move a pebble to v_{n-1}. If the pebble lies on a vertex v_i where $1 < i < n - 1$ then we can use this pebble to eventually land a pebble on v_n. Further, to move a pebble from v_1 to v_i costs 2^{i-1} if all pebbles are initially on v_1. Since $1 < i$ we save $2^{i-1} > 0$ pebbles to move one pebble to v_i. This pebble can then be used to pebble v_{n-1} and thus we need fewer than 2^{n-1} pebbles to pebble v_{n-1}. Therefore we do not need $\left\lfloor \frac{1}{3}(2^n - 1) \right\rfloor$ pebbles. Thus any initial configuration such that not every pebble is placed on an end vertex requires fewer than $\sigma(P_n)$ pebbles. □

2.2 Cycles

In order to find the covering cover pebbling number for a cycle we have two major steps. First we find the minimum number of pebbles needed to pebble a cover of a cycle if all the pebbles are initially placed on the same vertex. We do this by showing first that from such a distribution the cheapest cover is a minimum cover and then we calculate the cost for each of two possible minimum covers to identify which is cheapest. The second major step is to show that any we can pebble a cover of a cycle from any initial configuration of one fewer pebbles than the number we found in the first step as long as there are pebbles on at least two vertices. We do this by developing the notion of what we call a (u, v)-split to divide our cycle into two internally disjoint paths. We then justify that two such paths exist which we can pebble from our initial configuration to show that a cover of our cycle can be pebbled. This allows us to conclude that the covering cover pebbling number for a cycle of even length is the value we calculated in our first step since there is an initial configuration requiring that many pebbles and that number of pebbles is enough to pebble a cover from any other initial configuration.

Lemma 2.1. *The cheapest cover of a cycle C_n to pebble from a distribution where all pebbles are initially placed on one vertex is a minimum cover.*

Proof. Let C_n be a cycle with a starting vertex u, upon which all pebbles are placed. First pebble a cover of C_n with more than $\lceil n/2 \rceil$ vertices. Then at least one of the following two cases occurs. Either there are three vertices in a row which are pebbled or there are two pairs of adjacent vertices which have been pebbled. In the event of the first case we can simply not pebble the center vertex of the three vertices in a row and we will reduce the pebbling cost by at least one and reduce the number of vertices pebbled. Notice we still have a valid cover of C_n because the two edges covered by the middle vertex are also covered by the two vertices on either side of said vertex.

In the event of the second case there must be a pair of vertices closest to u on the left (call the closer of the two vertices a) and a pair of vertices closest to u on the right (call the closer of the two b). Further suppose that without loss of generality $d(u, a) = p$ and $d(u, b) = q$ and that $p \geq q$. There are two cases we need to consider here. Case 1: we will suppose that the shortest path from u to a does not include b. In Figure 9 we provide an example of what such a cover might look like on C_{18}. The white vertices denote vertices in the cover and the black ones are vertices which are not in the cover. Note that the only information we have for the (a, b)-path that includes v is that the neighbors of a and b each are in the cover. Although all the other vertices of this path are black in Figure 9, they may or may not be in the cover. Now we will label the vertices between a and u starting with a as $a_1, a_2, a_3, \ldots, a_p$ and the vertices between b and u starting with b as $b_1, b_2, b_3, \ldots, b_q$. Then the full (a, b)-path that includes u would be labeled as $a_1, a_2, \ldots, a_p, u, b_q, \ldots, b_2, b_1$. Since a and b are the closest vertices on either side of u who also have a neighbor pebbled, by assumption, it follows that every vertex we pebble between a and b does not have pebbled neighbors. Thus between a and b the vertices are alternating pebbled and not pebbled (i.e. a_1

is pebbled but a_2 is not and so on) because we have pebbled a cover of C_n and there are no adjacent pebbled vertices between a and b. Then instead of pebbling a_1 we can pebble a_2 and we still have a valid cover pebbled. Further, instead of pebbling a_2 we can pebble a_3 and so on. Ultimately this will conclude with pebbling b_2 and resulting in three in a row being pebbled, namely b_1, b_2, and the other neighbor of b_1. Thus, we find ourselves in the first case again. Further, the overall cost of this new pebbling compared to the original is less. Notice that each vertex labeled with a 'b' costs more than the one we did not pebble. However this increase in cost is balanced by the decrease of each vertex labeled with an a. Further we are not pebbling either a or b, so the net cost is less.

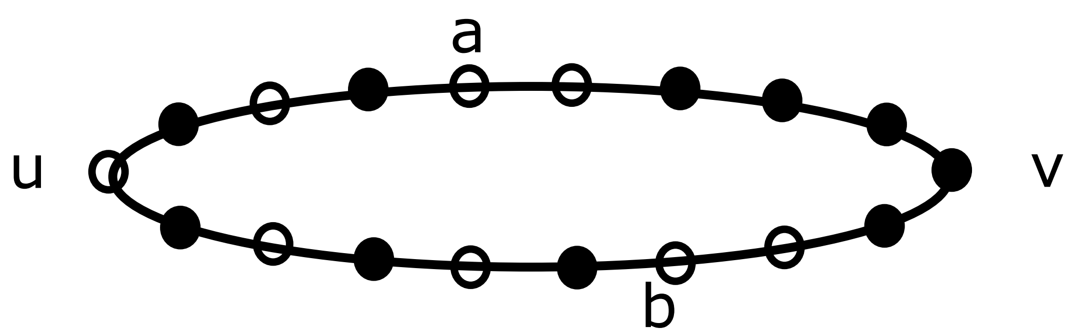

Figure 9: An example cover of C_{18}

Case 2: now suppose that the shortest path form u to a does include b. In Figure 10 we show what such a cover might look like on C_{18}. Here the white vertices represent vertices in the cover and the black vertices are vertics which are not in the cover. Note that the only information we have about the (a, b)-path which does not include u or v is that the neighbor of b is in the cover. Although the rest of the internal vertices are black in the Figure, they may or may not be in the cover.

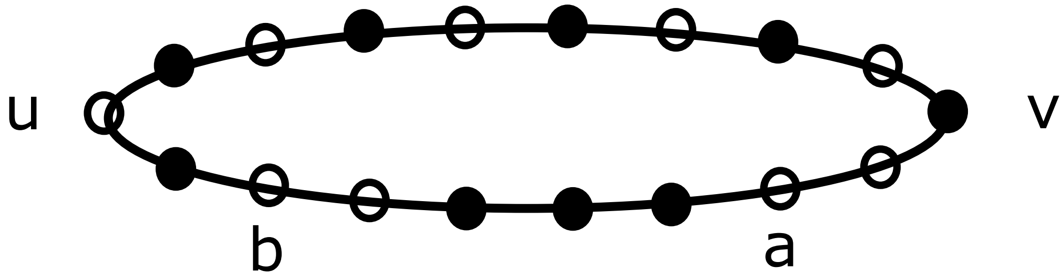

Figure 10: An example cover of C_{18}

Next, we will label the vertices from a to u, starting with $a = a_1$ as $a_1, a_2, a_3, \ldots, a_{p-1}$. Notice that $b = a_i$ for some i. Further since a_1 is pebbled but a_2 is not we can pebble a_2 instead of pebbling a_1. Note that if a_2 had been pebbled then we would have three vertices in a row, which is covered in case 1. This will still be a cover of C_n since by not pebbling a_1 we risk not covering the edge between a_1 and a_2, but by pebbling a_2 we resolve this. Further the cost of pebbling a_2 is half the cost of pebbling a_1 from u. Since a_2 was not pebbled in our original cover, a_3 must have been so our new cover has two in a row again. We can repeat this process until we pebble a_i and find a situation where a_i, a_{i+1}, and a_{i+2} are all pebbled. This must happen at some point because b and the neighbor of b are both pebbled. Thus we have found a new cover which costs less than the original cover and has three in a row. We can not pebble the middle cover, as in the first case, and we have found a cover which costs less than the original and uses fewer vertices. Therefore, in all cases we have found a way to pebble a cover C_n with fewer pebbles and fewer vertices. It follows that the cheapest cover to pebble is a minimum cover. □

Lemma 2.2. *The cheapest cover for a cycle C_n with n even for an initial configuration of all pebbles on one vertex u, is the minimum cover which does not include v, the vertex of distance $n/2$ from u.*

Proof. First consider $n \equiv 0 \pmod 4$. Suppose that u is a vertex of C_n upon which all pebbles are placed. We know from Lemma 2.1 that the cheapest cover of C_n that we could pebble is a minimum cover, of which there are two: the cover of $n/2$ vertices including v and the cover of $n/2$ vertices that does not include v. If we choose to pebble the cover with v we will need

$$p = 2^0 + 2(2^2 + 2^4 + \ldots + 2^{n/2-2}) + 2^{n/2} = 1 + 2^{n/2} + 2\sum_{i=1}^{n/4-1} 2^{2i} = \frac{1}{3}(2^{2+n/2} - 1) + 2^{n/2}$$

pebbles.

If we choose to pebble the cover without v we will need

$$q = 2(2^1 + 2^3 + \ldots + 2^{n/2-1}) = \sum_{i=1}^{n/4} 2^{2i} = \frac{4}{3}(2^{n/2} - 1)$$

pebbles. Then by comparing the two we see that the number of pebbles needed to pebble the cover containing v is always more than the number of pebbles needed to pebble the cover that does not contain v. Thus we see that q is a sufficient number of pebbles to pebble a cover of C_n when all pebbles are placed on a single vertex.

Next consider $n \equiv 2 \pmod 4$ and let u be the vertex upon which all pebbles are initially placed. Again from Lemma 2.1 we know the cheapest cover we can pebble from such a position is a minimum cover and again we see there are two possible minimum covers, the one which includes u and the one which does not. To pebble the cover without v we need

$$p = 2^0 + 2(2^2 + 2^4 + \ldots + 2^{(n-2)/2}) = 1 + 2\sum_{i=1}^{(n-2)/4} 2^{2i} = \frac{1}{3}(2^{2+n/2} - 5)$$

pebbles, while to pebble the cover with v we need

$$q = 2(2^1 + 2^3 + \ldots + 2^{(n-4)/2}) + 2^{n/2} = 2^{n/2} + \sum_{i=1}^{(n-2)/4} 2^{2i} = \frac{1}{3}(5(2^{n/2}) - 4)$$

pebbles. Here we can see that $q > p$ so p is a sufficient number of pebbles to pebble a cover of C_n when all pebbles are placed on a single vertex. In both cases we conclude that the cheapest cover we can pebble when starting with all pebbles on the same vertex is the cover which does not include v. $\square$

Here we introduce two notions which will be useful in determining the covering cover pebbling number of a cycle. First, we introduce the notion of a (u, v)*-split*. For C_n, a cycle on an even number of vertices a (u, v)*-split* is a pair of vertices u and v such that the distance between them is $n/2$. Observe that this split divides C_n into two internally disjoint (u, v)-paths of equal length, which we will refer to as P_A and P_B. In Figure 11 we show an example of a (u, v)-split on C_8. The dashed line is not part of the graph, but merely shows the dividing line between u and v. Second, we introduce the notion of allocating pebbles. We say that a pebble is *allocated to* $P(A)$ if the vertex it is initially placed on under ϕ is an internal vertex of P_A. It can also be *allocated to* P_A if it is initially placed on u or v under ϕ but if this is the case then it may be allocated to either P_A or P_B.

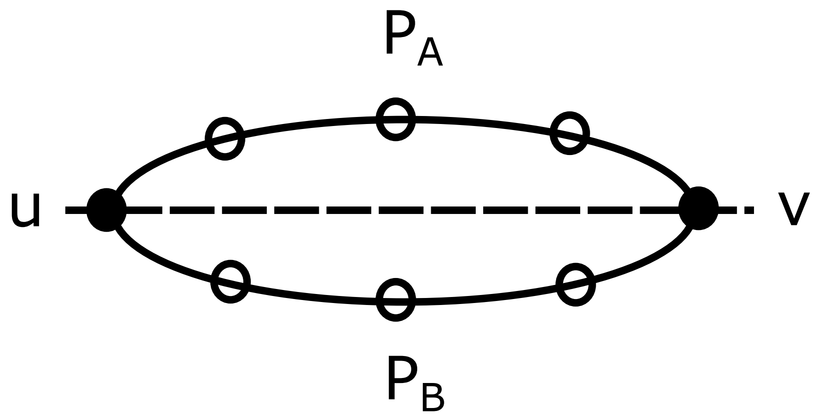

Figure 11: (u, v)-split on C_8

Lemma 2.3. *Let C_n be a cycle where $n \equiv 0 \pmod 2$ and let ϕ be a distribution of k pebbles onto C_n such that $k \equiv 0 \pmod 2$. Then C_n has a (u, v)-split such that precisely half of the vertices distributed onto C_n under ϕ can be allocated to P_A and the other half can be allocated to P_B.*

Proof. We prove this lemma via induction on the number of pebbles in ϕ. Let ϕ be an initial configuration of two pebbles on C_n, where $n \equiv 0 \pmod 2$. If both pebbles are on the same vertex then we can assign that vertex to be u and we have a (u, v)-split in which we can allocate one pebble to P_A and the other to P_B. If the pebbles are placed on different vertices we can select one vertex with a pebble on it and label it u. Then under the resulting (u, v)-split the other pebble must be internal to P_A or P_B or it is on v. In any of these cases we can allocate the pebble on u to whichever path the other pebble is not allocated to.

We assume that for all quantities of pebbles k where k is even and $k \leq s$ there is a (u, v)-split on C_n such that we can allocate $k/2$ pebbles to both P_A and P_B. Now let ϕ be a distribution of $s + 2$ pebbles onto C_n. If there are two vertices, a and b, that both have a pebble on them such that the distance between a and b is $n/2$, then we remove one pebble from a and one from b. This reduces the number of pebbles on C_n to n so there is a (u, v)-split of C_n. Then when we add the two removed pebbles back, one on a and one on b, we see that either a is internal to P_A and b is internal to P_B or vice-versa, or a and b are u and v. In the first case the (u, v)-split on C_n still preserves the property that P_A and P_B each can be allocated half the pebbles distributed on C_n under ϕ. In the second case we can choose which path P_A or P_B to allocate the pebbles we added back so we can still maintain the desired property.

Now assume that there is no pair of vertices with distance $n/2$ that each have a pebble on them. Then we remove two pebbles from C_n. Again we are left with only s pebbles on C_n and thus we have a (u, v)-split. We then return the two removed pebbles to C_n. If one of the pebbles is internal to P_A and the other to P_B then we still have the desired property. If one of the pebbles is returned to either u or v then we can simply allocate it to whichever path the other pebble is not allocated to, maintaining the desired property. So assume that both pebbles, when returned, are internal to one path and without loss of generality say that this path is P_A. By our assumption we know that at least one of u or v has no pebbles on it, since they are at distance $n/2$. Further we see that if one, say u, has pebbles on it and any of those pebbles were allocated to P_A before returning the pebbles, then we can allocate one of those to P_B and we have maintained the desired property. So we can assume that if there are any pebbles on u or v, again we will just say u without loss of generality, then they must all be allocated to P_B before we return the two pebbles.

Now we label the vertices in C_n adjacent to u with $u_{A,1}$ and $u_{B,1}$ and the vertices adjacent to v with $v_{A,1}$ and $v_{B,1}$ depending on whether the vertex is internal to P_A or P_B. Further, for notations sake we use $A(x)$ to denote the number of pebbles from x allocated to P_A and $B(x)$ to denote the number of pebbles from x allocated to P_B. Suppose $A(u_{A,1}) > B(v) = 0$. Then we can use the $(u_{A,1}, v_{B,1})$-split to achieve the desired property since this split will not change the number of pebbles allocated to P_A since $B(v) = 0$. Further since $A(u_{A,1}) > B(v) = 0$ we can allocate one more pebble to P_B and thus attain our desired property. A similar argument can be made if $A(v_{A,1}) > B(u)$. If $A(u_{A,1}) < B(v)$ and $A(v_{A,1}) < B(v)$ then we see that $B(u) > 0$ and $B(v) > 0$. Thus both u and v have a pebble under ϕ, which contradicts our assumption that no pair of vertices at distance $n/2$ have pebbles on them.

Assume, without loss of generality, that $A(u_{A,1}) = B(v) = 0$. Then we can use the $(u_{A,1}, v_{B,1})$-split and we see that P_A still has 2 more pebbles than P_B. Then we have a new $(u_{A,1}, v_{B,1})$-split where P_A has 2 more pebbles than P_B. Then if $A(u_{A,2}) = B(v_{B,1})$ we can use the next split. If this continues eventually we find a split with a vertex from which we removed and then replaced a pebble. At this point we can allocate the replaced pebble to P_B. When we removed the two pebbles the number of pebbles on P_A was equal to the number on P_B and when we replaced them we allocated one pebble to each path. Thus both paths have equal numbers of pebbles. If we assume that at some point $B(v_{B,i}) > A(u_{A,i+1})$ then we cannot use that split. So we return to the (u, v)-split we started with and consider splits in the other direction. If we are able to find a split with a vertex from which we removed and replaced a pebble then, by the above logic, we find a split where P_A and P_B have the same number of pebbles. If not, then we find some $u_{B,j}$ so that $B(u_{B,j}) > A(v_{A,j+1})$. Then we remove a pebble from $v_{B,i}$ and a pebble from $u_{B,j}$. We have a total of n pebbles so there must be an (a, b)-split so that there are an equal number of pebbles on P_A and P_B. Further we know that $v_{B,i}$ and $u_{B,j}$ cannot be on the same path unless one is either a or b in our split. We know this because when these two were in the same path that path had two fewer pebbles than the other path without removal of the pebbles from these vertices. Thus with removal any path with these two must have even fewer than the other path. But since after removal of pebbles from $v_{B,i}$ and $u_{B,j}$ we are able to find an (a, b)-split where P_A and P_B have equal numbers of pebbles $v_{B,i}$ and $u_{B,j}$ must be on different paths unless one is a or b. Then we replace the pebbles and, since $v_{B,i}$ and $u_{B,j}$ are on different paths, or one is an end point of the paths, the replaced vertices are allocated one to P_A and one to P_B and we have attained the desired property of a split where both paths have equal numbers of vertices.

Therefore we have shown that we can always find a (u, v)-split such that P_A and P_B have an equal number of pebbles allocated to them for any even number of pebbles distributed on C_n, where n is even.

$\square$

Lemma 2.4. *It is possible to pebble a cover of C_n from any initial configuration ϕ which distributes $(4/3)(2^{n/2} - 1) - 1$ pebbles onto C_n where $n \equiv 0 \pmod 4$ or $(1/3)(2^{n/2+2} - 5) - 1$ pebbles onto C_n where $n \equiv 2 \pmod 4$ such that there are pebbles on at least two vertices of C_n.*

Proof. First consider C_n with $n \equiv 0 \pmod 4$ and let ϕ be a distribution of $4/3(2^{n/2} - 1) - 2$ pebbles onto C_n such that at least two vertices of C_n have pebbles. We see that $4/3(2^{n/2} - 1) - 2$ is even and thus there exists a (u, v)-split on C_n by Lemma 2.3. Then there are internally disjoint paths P_A and P_B, with $1/3(2^{n/2+1} - 2) - 1$ pebbles allocated each. Further we know that $\sigma(P_{n/2+1}) = 1/3(2^{n/2+1} - 2)$ from Theorem 1.2. We know that at least one path, say P_A, has more than one vertex with pebbles under ϕ and from [6] we know we need $1/3(2^{n/2+1} - 2)$ pebbles if all pebbles are on an end point. It follows from Theorem 2.1 that P_A requires fewer than $1/3(2^{n/2+1} - 2)$ pebbles and given that it has $1/3(2^{n/2+1} - 2) - 1$, which is one less, we are able to pebble a cover of P_A. Now if P_B has pebbles on more than one vertex, or if it has all of its pebbles on an internal vertex, then just like P_A, we can pebble a cover of P_B by Theorem 2.1. So suppose that all pebbles allocated to P_B are located on u, without loss of generality. Then add a pebble to any vertex of C_n. This new distribution of $4/3(2^{n/2}-1)-1$ pebbles onto C_n will be called ϕ'. Now if the new pebble can be allocated to P_B then we have $1/3(2^{n/2+1}-2)$ pebbles allocated to P_B so we can pebble a cover of P_B by Theorem 1.2.

Suppose that the new pebble is internal to P_A. Then if P_A has a pebble allocated to it on either u or v then we can allocate that to P_B instead. By the above argument we can still pebble a cover of P_A and with the additional pebble we can also pebble a cover of P_B.

Finally, if the pebble we added is internal to P_A and there are no pebbles allocated to P_A from u or v then we consider the $(u_{A,1}, v_{B,1})$-split. We will refer to the $(u_{A,1}, v_{B,1})$-path which has most of its vertices in common with P_A as P'_A and the $(u_{A,1}, v_{B,1})$-path which has most of its vertices in common with P_B as P'_B. Notice that P_A and P'_A have the same number of pebbles allocated since the only vertex in P_A that is not in P'_A is u, which had no pebbles allocated to P_A by assumption. Thus we can pebble a cover of P'_A, since it has $1/3(2^{n/2+1}-2)$ pebbles allocated to it. Further, all of the pebbles allocated to P'_B are still on u, by assumption, and this is now an internal vertex of P'_B. Thus, we have $1/3(2^{n/2+1}-2)-1$ pebbles on an internal vertex of P'_B, which is enough to pebble a cover of P'_B, by Theorem 2.1. Therefore, we can pebble a cover of C_n from any distribution of $4/3(2^{n/2}-1)-1$ pebbles onto C_n where $n \equiv 0 \pmod 4$ and there are pebbles on at least two vertices.

Next consider C_n where $n \equiv 2 \pmod 4$ and let ϕ be a distribution of $1/3(2^{2+n/2}-5)-1$ pebbles onto C_n such that there are pebbles on at least two vertices. Since $1/3(2^{2+n/2}-5)-1$ is even we know from Lemma 2.3 that there is a (u,v)-split of C_n so that P_A and P_B can both be allocated $1/3(2^{n/2+1}-4)$ pebbles. Further we know that at least one path, say P_A, has pebbles allocated to it from more than one vertex. Thus this path, which needs $1/3(2^{n/2+1}-1)$ pebbles if all are on an end point, has a cover which includes either u or v that can be covered [6]. Now P_B has one less pebble than it may need to pebble a cover but if P_A pebbles either u or v then P_B has a cover which includes that vertex and thus needs one fewer pebble since that vertex has already been pebbled from P_A.

Thus we conclude that it is possible to pebble a cover of C_n from any initial configuration ϕ which distributes $p-1$ pebbles onto C_n such that at least two vertices receive pebbles, where $n \equiv 0 \pmod 2$ and p is the number of pebbles needed to pebble a cover of C_n if all pebbles are initially placed on a single vertex.

□

Theorem 2.2. *Let C_n be a cycle of even length. Then*

$$\sigma(C_n) = \begin{cases} \frac{4}{3}(2^{\frac{n}{2}} - 1) & \textit{when } n \equiv 0 \pmod 4 \\ \frac{1}{3}(2^{\frac{n}{2}+2} - 5) & \textit{when } n \equiv 2 \pmod 4 \end{cases}$$

Proof. From Lemma 2.2 we know that if all pebbles are placed on a single vertex then we need at least $p = \frac{4}{3}(2^{\frac{n}{2}} - 1)$ pebbles to pebble a cover of C_n if $n \equiv 0 \pmod 4$ and $q = \frac{1}{3}(2^{\frac{n}{2}+2} - 5)$ pebbles to pebble a cover of C_n if $n \equiv 2 \pmod 4$. From Lemma 2.4 we know that it is possible to pebble a cover of C_n from any initial configuration ϕ which distributes $p-1$ pebbles onto C_n such that at least two vertices receive pebbles, where $n \equiv 0 \pmod 4$. We also know that it is possible to pebble a cover of C_n from any initial configuration ϕ which distributes $q-1$ pebbles onto C_n such that at least two vertices receive pebbles, where $n \equiv 2 \pmod 4$. Thus we have shown that when $n \equiv 0 \pmod 4$ we can pebble a cover from any initial configuration of p pebbles and when $n \equiv 2 \pmod 4$ we can pebble a cover from any initial configuration of q pebbles. Further there is a configuration of pebbles onto C_n where $n \equiv 0 \pmod 4$ where fewer than p pebbles is insufficient to pebble a cover. Similarly there is a configuration of pebbles onto C_n where $n \equiv 2 \pmod 4$ for which fewer than q pebbles is insufficient to pebble a cover. Therefore, we have shown the covering cover pebbling number σ is as claimed for cycles of even length. □

2.3 Ladders

We can then apply these results on paths and cycles to ladders. A *ladder* is a graph $\mathcal{L}_n = P_n \square K_2$ where P_n is a path of length n and K_2 is a complete graph on two vertices (also a path of length 1). We give an example of $\mathcal{L}_5$ in Figure 12.

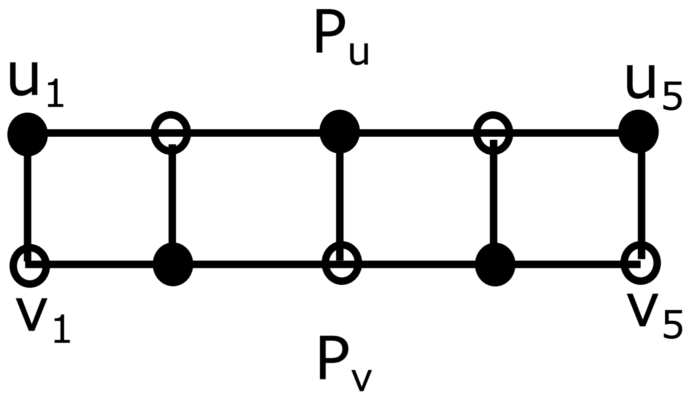

Figure 12: Ladder $\mathcal{L}_5$

Theorem 2.3. *The covering cover pebbling number for a ladder is precisely the covering cover pebbling number for a cycle twice the length of its defining path. Notationally* $\sigma(\mathcal{L}_n) = \sigma(C_{2n})$.

Proof. Let $\mathcal{L}_n$ be the ladder with a defining paths $P_u = u_1, u_2, \ldots, u_n$ and $P_v = v_1, v_2, \ldots, v_n$. First note that $\mathcal{L}_n$ has a Hamiltonian cycle, H, consisting of P_u, P_v, and edges u_1v_1 and u_nv_n. This Hamiltonian cycle is of even length so it is 2-colorable (i.e. we can label the vertices alternating with '0' and '1' around H). In Figure 12 we see an example of coloring the vertices of $\mathcal{L}_5$ with white and black. Now consider an edge of $\mathcal{L}_n$ that is not in H. Such an edge is of the form u_iv_i. Then since we alternate zeros and ones around H we know that u_1 and v_1 are labeled differently. Then it follows that u_i and v_i have different labels as well. Therefore, we can conclude that every edge of $\mathcal{L}_n$ is between a vertex labeled zero and a vertex labeled one given this labeling of H. By pebbling all of the vertices of the same label we then pebble a cover of $\mathcal{L}_n$. Such a cover is also a minimum cover of H.

Observe that in order to pebble a vertex u_i from u_1 we need 2^{i-1} pebbles. This is true because the shortest distance is following H. Similarly, to pebble v_j from u_1 we need $2(2^{j-1})$. We cannot reach v_j with less and with $2(2^{j-1})$ we can simply traverse H to reach v_j. Thus if we place all pebbles at u_1 (or really any corner vertex of $\mathcal{L}_n$) we need the same number of pebbles as to pebble C_{2n} as the cheapest way to pebble any vertex of $\mathcal{L}_n$ is to follow H, which is a cycle of length $2n$. Thus $\sigma(\mathcal{L}_n) \geq \sigma(C_{2n})$.

Now suppose that we have a configuration of $\sigma(C_{2n}) - 1$ pebbles onto $\mathcal{L}_n$. If this configuration has pebbles on more than one vertex then by Lemma 2.4 we can pebble a cover of $\mathcal{L}_n$. If the new configuration has all pebbles placed onto the same vertex, u_i but that vertex is not a corner vertex, then there is a edge in $\mathcal{L}_n$ from u_i to a vertex that is not u_{i-1} or u_{i+1}, namely v_i. This means we need fewer than $\sigma(C_{2n})$ pebbles to pebble a cover of $\mathcal{L}$.

Therefore $\sigma(\mathcal{L}_n) = \sigma(C_{2n})$ and further the most expensive initial configuration for pebbling a cover of $\mathcal{L}_n$ is one in which all pebbles are placed on the same corner vertex. $\square$

2.4 Products of Paths

In order to determine the covering cover pebbling number for a product of paths we prove two theorems that, combined, give us our desired result. First we show that the worst case scenario for a product of paths is the initial configuration with all pebbles placed on one corner vertex. A *worst case scenario* for a graph G has two properties. It is an initial configuration from which a cover of G can be pebbled but the removal of any pebble will make it impossible to pebble a cover of G. Further, of all initial configurations with the first property, a worst case scenario uses at least as many pebbles as any other. Then we show that from a worst case scenario the cheapest cover which we can pebble is the minimum cover which does not include the corner vertex diametrically opposite the vertex upon which all pebbles are initially placed. From this we can conclude that the covering cover pebbling number must be the number of pebbles needed to pebble the minimum cover which does not include the opposite vertex of the starting vertex from the initial configuration of a worst case scenario, where all pebbles are initially placed on the same corner vertex.

Lemma 2.5. *For any initial configuration ϕ on $G = P_n \Box P_m$ with exactly enough pebbles to reach a winning configuration, there exists an initial configuration ϕ', using the same number of pebbles as ϕ, for which there is no reachable winning configuration or there is a $P_n \Box P_i$ subgraph of G with enough pebbles on it to pebble a cover of itself under ϕ'. Further, if there is a reachable winning configuration then ϕ' distributes precisely enough pebbles onto G to reach this winning configuration.*

Proof. Let ϕ be a distribution of pebbles onto $G = P_n \Box P_m$ such that there is some cover C which can be pebbled from ϕ but there is no cover which can be pebbled that does not require every pebble distributed under ϕ. Further assume that ϕ is not a distribution with every pebble placed on the same corner vertex. Note that if ϕ were such an initial configuration then the result is trivial. We alter ϕ to create ϕ', another initial configuration with the same number of pebbles as ϕ, as follows. For $1 \leq i \leq m+1$ if there is a pebble on vertex (i, j) that can be used to pebble a vertex in row $k > i$ then relocate that pebble to vertex $(1, j)$. Repeat the relocation process until there is no cover that can be pebbled or there is a cover C' that can be pebbled such that no pebble in row i is used to pebble a vertex in row k where $k > i$ for $1 < i \leq m + 1$. If there is such a cover C' then we will use the configuration after the relocation as a new initial configuration and call it ϕ'. Further we observe that the number of pebbles needed to pebble C' from ϕ' is the same as the number of pebbles needed to pebble C from ϕ. This is true because at each iteration we move pebbles to row one. This raises the cost of pebbling vertices in higher numbered rows with the same pebbles that were used in the previous iteration. Further, if at any iteration we use a different set of pebbles to pebble a vertex that was originally pebbled (or its neighbor was pebbled) by pebbles we have moved to row one they must come from a lower numbered row than that vertex is in. Any pebbles in a higher numbered row are needed to pebble something that cannot be reached by the pebbles we move to the first row.

Suppose there is a vertex $u \in C'$ in row i that is pebbled by at least one pebble in row $j > i$ under ϕ'. We will use B to refer to the set of pebbles we use to pebble u under ϕ'. Further suppose that we could pebble u using pebbles we moved to the first row. For brevity we will refer to this set of relocated pebbles as A. Now these pebbles were used to pebble something that is not u under ϕ, because if they were used to pebble u under ϕ then we would need more pebbles after the relocation. Call one of these vertices v and say it is in row k. We know that if $k \geq i$ then there is no way we can pebble u from the first row because there were exactly enough to get to row i and k is further away. If $i < k$ then, under ϕ we should have pebbled u with A and v with B. Overall, we will need fewer of the pebbles in A and B to pebble u and v respectively, since u is closer to the vertices we got A from than v is. This contradicts our assumption that we had exactly enough pebbles under ϕ to pebble C.

From ϕ' we create a new initial configuration ϕ'' by repeating a similar relocation process, where if a pebble on vertex (i, j) is used to pebble a vertex in row k and $k < i$ then we relocate this pebble to $(m + 1, j)$, until there is no cover that can be pebbled or there is a cover C'' that can be pebbled so that no pebble in row i is used to pebble a vertex in row k where $k > i$ for $1 \leq i < m + 1$. Since this process is identical to the original relocation process if we relabel the rows from top to bottom the results above hold and pebbling C'' from ϕ'' costs the same as pebbling C from ϕ.

Note that there could be a way to pebble C'' from ϕ'' in which we use pebbles not in the first row to pebble a vertex in a higher numbered row. We will not have a pebble which is not in the $m + 1$ row used to pebble something in a lower numbered row because this would simply have another iteration of the second relocation process. Suppose that there is a pebble in row i, where $1 < i$, on vertex v that we might use to pebble a vertex u in row $j > i$. This in turn means we need to use pebbles from row 1 to pebble the vertex in row i originally pebbled with the pebble we are now using to pebble u from row i. It does not make sense to use pebbles from row $m + 1$ since to move a pebbles from row $m + 1$ to row i we move them past row j first. Thus we could save the pebbles needed to move a pebble from row i to row j and also a pebble from row j to row i but just pebbling u from $m + 1$ rather than use the pebble in row

i. If we assume we cannot do this then we must move pebbles from the first row to row i. The original way we pebbled C'' was to move pebbles from row 1 to row j but now we are moving pebbles from row 1 to row i and from i to row j. Notice that the distances we are moving pebbles is originally $j-1$ and now it is $j-i+i-1=j-1$. Thus the vertical distances that pebbles are moved is the same. The only way that the new way of moving pebbles could be cheaper is if the horizontal distance is shorter than the horizontal distance pebbles are moved in the original way of moving pebbles. However, horizontal distances are maintained by the relocation process, since we never move a pebble to a different column. Thus this distance must be the same as well. So such a modification of how we move pebbles cannot decrease the cost of pebbling C'' from ϕ''.

Now we observe the subgraph consisting of the top row or the subgraph consisting of the bottom row must have enough pebbles on it to pebble a cover of itself. If this were not the case then there would be pebbles from the bottom row that need to be moved to the top row and pebbles from the top row that need to be moved to the bottom row. However, the greatest distance a pebble can be moved in the top two rows is $n-1$, which is at most equal to the shortest distance that a pebble needs to be moved to get from the bottom row to the top (namely m). Therefore, for any pebble we move from the bottom row to the top we could instead use a pebble in the top row for at most the same cost. This implies that we must be able to pebble a cover of at least one of the top or bottom row using only pebbles from that row.

Thus, we have a new initial configuration ϕ'' which uses the same number of pebbles as ϕ, and this number is exactly enough to reach a winning configuration. Further the only way we cannot find such a new initial configuration is if the relocation processes terminate because there is no reachable cover. Therefor, we have proven our desired result. $\square$

Theorem 2.4. *The worst case scenario for pebbling a product of paths $P_n \square P_m$ is an initial configuration which places all pebbles on the same corner vertex.*

Proof. We know that for a graph $G = P_n \square P_2$, namely a ladder, that the worst case scenario is an initial configuration in which all pebbles are placed on a corner vertex of G, as shown in Theorem 2.3. So assume that for $P_n \square P_i$, where $2 \leq i \leq m$, the worst case scenario is that all pebbles are placed on a corner vertex and consider $G = P_n \square P_{m+1}$. Without loss of generality we can let $m + 1 \geq n$. Now let ϕ be a distribution of pebbles onto G such that there is some cover C which can be pebbled from ϕ but there is no cover which can be pebbled that does not require every pebble distributed under ϕ. Further assume that ϕ is not a distribution with every pebble placed on the same corner vertex. Now from Lemma 2.5 we know that there is an initial configuration ϕ using the same number of pebbles as ϕ from which either there is no cover which can be pebbled or there is a $P_n \square P_i$ subgraph of G (with $1 \leq i < m + 1$) with enough pebbles to cover itself. If there is no cover which can be reached from ϕ' then ϕ cannot have been a worst case scenario.

So suppose that there is a subgraph, $A = P_n \square P_i$, of G with enough pebbles to pebble a cover of itself, where $1 \leq i$. We will now proceed by cases. First, suppose that all of the pebbles needed to pebble A are on an outer corner p of A (and thereby also a corner vertex of G). If the pebbles needed to pebble the other portion of G, namely a $B = P_n \square P_{m+1-i}$ subgraph, are all on B and further located on an outer corner q then by relocating these pebbles to p we cannot have enough pebbles to pebble a cover of G. This is true because ϕ'' had exactly enough pebbles to pebble a cover of G, as shown above, and relocating these pebbles will move them further from vertices they need to be moved to, making it impossible to pebble them.

If the pebbles need to pebble B are all on B but not all on on a corner vertex then by relocating those pebbles to a corner vertex we see we do not have enough pebbles to pebble a cover of G by our inductive hypothesis. Finally if the pebbles needed to pebble a cover of B are not all on B then there must be some pebbles on A which are needed for B. These can be moved to p, if they are not already on p, which will increase cost. Further the pebbles which are on B can still be moved to q to increase cost (again by inductive hypothesis). Further, by the argument above we could relocate pebbles from

q to p and increase cost again. Note that we cannot have all of the vertices starting on p under ϕ''. The only way this could happen given our relocation algorithm is if every pebble either started on p, which did not happen by assumption, or if that all pebbles were placed on vertices in the same column as p and then moved to higher numbered rows under ϕ. Suppose we have such a configuration. Then all the pebbles in row i are used to pebble vertices in higher numbered rows and we need to pebble the vertices in row i with pebbles from lower numbered rows. Note that the cost to pebble a vertex u in row i from row j where $j < i$ is the cost of moving pebbles from row j to row i and then the cost of moving along row i to the desired vertex. Thus we can move the pebbles from row j to row i through pebbling moves and we will have enough sitting on the first vertex in row i to pebble u and to pebble whatever was pebble with the vertices in i. Then we can use one pebble that was in row i to help pebble u and use one of the pebbles we moved to row i to help pebble whatever was pebbled from row i. Thus we can use a pebble in row i to pebble something in the same row and as itself so we would not relocate this with the relocation algorithm. Thus if we have all of the pebbles needed to pebble a cover of A placed on p, but there are pebbles on B and thus not on p, then we can find an initial configuration which requires more pebbles than ϕ'' and thus more than ϕ to pebble a cover of G. Therefore ϕ could not have been a worst case scenario.

Second, suppose that all pebbles needed to pebble a cover of A are on A but not all of the pebbles are on p under ϕ''. Then by relocating these pebbles to p we increase the cost of pebbling A by our inductive hypothesis. There may be more pebbles on A and after the first such relocation we may be able to use some of these to still pebble a cover of A. But if we continue this relocation process to p eventually we will have either a configuration from which we cannot pebble a cover of A or we have reduced this case to the first case and can proceed from there. Either way we show that ϕ'' cannot be a worst case scenario and thus neither can ϕ.

Therefore, by induction, we conclude that the worst case scenario for any product of paths must be an initial configuration where all pebbles are placed on a corner. $\square$

Theorem 2.5. *For $P_n \square P_m$ the cheapest cover to pebble from the initial configuration of all pebbles on a corner vertex, p, is the minimum cover which does not include the opposite corner from p.*

Proof. Let ϕ be an initial configuration with all pebbles on the upper left corner of $G = P_n \square P_m$, call this vertex p. We know that G is bipartite. Thus we can label each vertex with either a 0 or 1 in such a way that every edge has an endpoint labeled 0 and an endpoint labeled 1. Now define C to be the set of vertices of G which all have the opposite label of vertex in the lower right corner. Notice that C is a cover of G. Further notice that C is a minimum cover of G as any fewer vertices would result in a pair of adjacent vertices which are not in the cover. Now let D be another cover of G. Then we define $f : C \to D$ as follows. If v is in C and also in D then $f(v) = v$. If v is in C but not in D then all of the neighbors of v must be in D. If v has a neighbor to the right, call it v_r then $f(v) = v_r$. Now v_r is one step further away from p than v is so it will cost twice as much to pebble v_r from p as it will cost to pebble v from p. If v has no neighbor to its right then it must have a neighbor below it, call that vertex v_d, and $f(v) = v_d$. Note that the only vertex with no neighbor to its right or below it is the lower right corner vertex of G, which is not in C. Now v_d, much like v_r is one step further away from p than v is so it will cost twice as much to pebble v_d from p as it will cost to pebble v from p. Observe that with this mapping every vertex in C is mapped to a vertex in D which at least as much to pebble. Further the only time more than one vertex in C is mapped to the same place in D precisely two vertices in C of the same cost are mapped to a vertex in D costing twice as much. Thus if u and v are in C and $f(u) = f(v)$ then the cost of pebbling u is the same as the cost of pebbling v and the cost of pebbling $f(u)$ is twice the cost of pebbling u so the net change in cost is 0. Therefore we have shown that the set of vertices mapped to in D from C must cost at least as much to pebble as C. Further, since D is not the same cover as C, there must be some vertex v in D which is not in C. Either this was mapped to under f, in which v cost more to pebble from p than $f^{-1}(v)$, or this vertex was not mapped to under f, in which case we need more pebbles in order to pebble this extra vertex.

In either case the cost of D is shown to be strictly greater than than cost of C. $\square$

Theorem 2.6. *The covering cover pebbling number for a product of paths $P_n \square P_m$ is given by*

$$\sigma(P_n \square P_m) = \begin{cases} \frac{4}{9}(2^m - 1)(2^n - 1) & \text{when } n, m \equiv 0 \pmod 2 \\ \frac{1}{9}(2^{2+m+n} - 5(2^m) - 5(2^n) + 4) & \text{when } n, m \equiv 1 \pmod 2 \\ \frac{1}{9}((2^{2+m} - 5)(2^n - 1) - 3m2^n) & \text{when } n \equiv 0 \text{ and } m \equiv 1 \pmod 2 \end{cases}$$

Proof. We have shown in Theorem 2.4 that for a product of paths $P_n \square P_m$ the initial configuration requiring the most pebbles to pebble a cover is the configuration with all pebbles on a corner vertex. Therefore, the number of pebbles needed to pebble a cover of $P_n \square P_m$ will be sufficient to pebble a cover of $P_n \square P_m$ from any initial configuration. Further we have shown in Theorem 2.5 that the cheapest cover of $P_n \square P_m$ that can be covered from the worst case scenario is the minimum cover of $P_n \square P_m$ which does not include the corner vertex opposite the starting vertex. Thus if m and n are both even we see that we need $2(2^1) + 4(2^3) + \ldots + m(2^{m-1}) + m(2^{m+1}) + m(2^{m+3} + \ldots + m(2^{n-1}) + (m-2)2^{n+1} + (m-4)2^{n+3} + \ldots + (2)2^{m+n-3}$. This is equivalent to $\sum_{i=1}^{m/2} 2i(2^{2i-1}) + \sum_{j=0}^{(n-m)/2-1} m(2^{m+2j+1}) + \sum_{k=1}^{m/2}(m-2k)2^{n+2k-1} = \frac{4}{9}(2^m - 1)(2^n - 1)$. If both m and n are odd then we need $2(2^1) + 4(2^3) + \ldots + (m-1)(2^{m-2}) + m(2^m) + m(2^{m+2} + \ldots + m(2^{n-2}) + \ldots + (2)2^{m+n-3}$ which is equivalent to $\sum_{i=1}^{(m-1)/2} 2i(2^{2i-1}) + \sum_{j=0}^{(n-m)/2-1} m(2^{m+2j}) + \sum_{k=0}^{(m-1)/2} (m-2k-1)2^{n+2k} = \frac{1}{9}(2^{2+m+n} - 5(2^m) - 5(2^n) + 4)$. If n is even and m is odd then we need $2^0 + 3(2^2) + 5(2^4) + \ldots + m(2^{m-1}) + m(2^{m+1}) + \ldots + m(2^{n-2}) + (m-1)2^n + \ldots + 2(2^{m+n-3})$ which is equivalent to $\sum_{i=0}^{(m-1)/2} (2i+1)(2^{2i}) + \sum_{j=(m+1)/2}^{(n-2)/2} m(2^{2j}) + \sum_{k=0}^{(m-1)/2} (m-2k-1)2^{n+2k} = \frac{1}{9}((2^{2+m} - 5)(2^n - 1) - 3m2^n)$. Thus we have shown that the covering cover pebbling number for a product of paths $P_n \square P_m$ is given by

$$\sigma(P_n \square P_m) = \begin{cases} \frac{4}{9}(2^m - 1)(2^n - 1) & \text{when } n, m \equiv 0 \pmod 2 \\ \frac{1}{9}(2^{2+m+n} - 5(2^m) - 5(2^n) + 4) & \text{when } n, m \equiv 1 \pmod 2 \\ \frac{1}{9}((2^{2+m} - 5)(2^n - 1) - 3m2^n) & \text{when } n \equiv 0 \text{ and } m \equiv 1 \pmod 2 \end{cases}$$

as desired.

□

3 Future Research

We would like to extend these results to other families of graphs with similar structure, such as $P_n \square C_m$ and $C_n \square C_m$. We believe that by applying the above techniques to these families of graphs we will be able to determine the covering cover pebbling numbers of them as well. At the very least we should be able to use the covering cover pebbling number of $P_n \square P_m$ as an upper bound for the covering cover pebbling number of both $P_n \square C_m$ and $C_n \square C_m$.

Another open question we have is for what types or families of graphs is a worst case scenario an initial configuration in which all pebbles are placed on the same vertex. We have shown this to be the case for cycles, and products of paths, and [path citation] showed it to be true for paths. We also strongly suspect that the worst case scenario for a Cartesian product of cycles and the Cartesian product of a path with a cycle is an initial configuration with all pebbles on one vertex, although we have not proven this as of yet. However, for many bipartite graphs, especially complete bipartite graphs, this is not the case. A star on 4 vertices is a very simple example. With one pebble on the central vertex, or two pebbles on any one vertex a cover can be obtained after at most one move. However one could place one pebble on each of two leaves and then no cover can be achieved. Thus the worst case scenario is clearly not all pebbles on one vertex. It can be shown by a similar argument that $K_{n,n}$ does not have this worst case scenario property and $K_{n,n}$ is vertex transitive. So we do know that vertex transitivity is not a condition for this worst case scenario property.

References

[1] F.R.K. Chung, *Pebbing in Hypercubes*, SIAM J.Discrete Math **2(4)** (1989), 467-472.

[2] Crull et. al., *The cover pebbling number of graphs*, J. Discrete Math **296** (2005), 15-23.

[3] S. Elledge, G Hurlbert, *An Application of Graph Pebbling to Zero-Sum Sequences in Abelian Groups*, Integers **5(1)** (2005), A17.

[4] G. Hurlbert, *A survey of graph pebbling*, Congressus Numerantium **139** (1999), 41-64.

[5] G. Hurlbert, B. Munyan, *Cover pebbling hypercubes*, Bull. Inst. Combin. Appl. **47** (2006), 7176.

[6] A. Lourdusamy, A. Punitha Tharani, *Covering Cover Pebbling Number*, Utilitas Mathematica **78** (2009), 41-54.

[7] A. Lourdusamy, T. Mathivanan, *Covering Cover Pebbling Number For Square of a Cycle*, JPRM **8** (2012), 102-105.

[8] A. Lourdusamy, T. Mathivanan, *Covering Cover Pebbling Number For Square of a Path*, Gulf Journal of Mathematics **2(2)** (2014), 83-87.

[9] A. Lourdusamy et. al., *Covering Cover Pebbling Number for Even Cycle Lollipop*, Gen. Math. Notes **5(2)** (2011), 24-41.